Motion Conversion based on the Musculoskeletal System

Motion Conversion based on the Musculoskeletal System

Taku Komura[*] **Yoshihisa Shinagawa**[†] **Toshikazu EBISUZAKI**[‡]

[*] Image Information Division, Advanced Computing Center, RIKEN
2-1, Hirosawa, Wako-shi, Saitama 351-0198 Japan
e-mail: kohmura@atlas.riken.go.jp
[†] Department of Information Science, Faculty of Science, The University of Tokyo
7-3-1 Hongo, Bunkyo-ku Tokyo 113-0033 Japan
e-mail: sinagawa@is.s.u-tokyo.ac.jp
[‡]Computational Science Division, Advanced Computing Center, RIKEN
2-1, Hirosawa, Wako-shi, Saitama 351-0198 Japan
e-mail: ebisu@atlas.riken.go.jp

There is an increasing demand for techniques to accurately edit, convert, and retarget captured human motion data. Various methods have been proposed for such applications. However, there are still limitations which need to be overcome: first, these techniques often reduce the realism of the motion as the difference between the original motion and the resulting motion increases, and second, changes to simulate physiological effects cannot be achieved. These limitations are due to the fact that the the anatomical structure of the human body is not taken into account. While editing real human motion, even when physical laws are followed, if the dynamics of the human body are ignored, the resulting motion can become unrealistic. Further, if animators want to add physiological effects to the motion, such as the effects of fatigue or injuries, then muscle dynamics are an essential factor that must be taken into account.

In this paper, we propose a method for converting physical and physiological motion based on space-time constraints. The method involves a combination of musculoskeletal models and inverse dynamics to avoid the limitations of previous methods. This method makes it possible to apply physiological and dynamic conversions to captured motion, while also maintaining physical laws. **Keywords.** *animation, human body simulation, physically based animation*

1 Introduction

In computer graphics and animation, motion-capturing is the most practical way to obtain realistic human motion data. Since motion-capturing requires special and expensive equipment, animators generally try to use kinematic trajectories that are already available in databases. If an animator cannot find the exact desired motion, it is necessary to apply some kind of editing and modification to the available data. Even if an exact motion is found, the motion often needs to be adapted to the size of the body performing the

action. For this reason, there is an increasing demand to edit, convert, and retarget real human motion data.

Various methods have been proposed to edit real human motion data. Such methods can be divided into two groups: kinematic methods and dynamic methods.

The former method only converts the trajectories of the body kinematically, without taking into account dynamics. Therefore, as the difference between the original motion and the edited motion increases, the more likely will the resulting motion look unnatural.

The latter method takes into account dynamics for converting or modifying the motion. Since physical laws are followed with this method, the resulting motion generally looks more natural than that edited by the kinematic method. However, since the internal structure of the human body is ignored, even though the motion may be correct according to physical laws, it can appear physiologically unnatural.

Both methods also have the limitation that adding a physiological effect to the motion is difficult. For example, it is quite difficult to create a fatigued motion from ordinary motion by either kinematic or dynamic editing. It is also difficult to convert a motion to a similar motion made by an injured body. Neither the kinematic nor dynamic method takes into account the information about the internal structure of the body. Even if such conversion or editing were attempted, it would require a great amount of trial-and-error effort in tuning number of parameters.

In this paper, we propose a method that uses a musculoskeletal human body model and inverse dynamics to allow such conversion and increase the realism of the edited motion. As inverse dynamics and a musculoskeletal system are used together, it is also possible to evaluate whether a given motion can be achieved by the human body.

Our method is based on space-time constraints. It is known that the space-time constraints method is not suitable to models with many degrees of freedom such as the human bodies. However, initialization of the optimization process from real motion data, and generous feasibility optimization criteria, enable an animator to obtain natural physiologically feasible motion with our method.

By converting a motion while taking into account dynamics and the physiological feasibility of the motion, it is possible to add effects such as fatigue and injury to the motion quite easily, without any parameter tuning or trial and error. Since dynamics are taken into account, alternation of physical parameters such as gravity or external force applied to the body is also possible.

Even though the initial implementation of the musculoskeletal human body model is hard work, once it has been prepared, all further calculations can be done automatically.

2 Previous Work

The techniques to create human animation can be divided into three groups: methods based on kinematics, physical simulation, and space-time constraints.

Inverse kinematics, a method to determine the rest of the joint angles from the position of the end effectors, is a major method to create human animation kinematically. Zhao and Badler [39] have proposed a numerical method for determining the posture of a human figure which can also be applied to any tree-structured system. The problem with inverse kinematics is that no dynamics are taken into account.

To make animation more realistic, methods based on physical simulation have been developed (e.g., [1, 2, 33]). A new problem that arises when controlling a human body model in a physical environment is that the animator must describe the changes in the torque and force applied to the model, instead of the kinematic trajectories. This is a difficult task because the effect of changing each dynamic parameter is not obvious. To resolve this problem, a method known as proportional-derivative (PD) control has been developed by many researchers. This approach has been used to simulate gaits [20, 31] and athletic movements such as running, jumping and cycling [16], and diving [37].

Another approach is by spacetime constraints [36] which treats a motion synthesis problem by constrained optimization. Keyframe animation techniques based on this method [5, 23, 18, 19] have been developed.

A different way to obtain human motion data is to use real human data. Motion capturing devices are used in such a case. If the animator cannot find the desired motion, it is necessary to apply some kind of editing or modification to the available data. Even if an exact motion can be found, if the size of the body performing the action does not match that of the character, the motion must somehow be adapted to the character which the animator wants to control. For this reason, there is an increasing demand to edit, convert, and retarget real human motion data. Previous techniques to create human animation automatically have often utilized to achieve this.

Various methods have been proposed to kinematically edit motion-captured data. The techniques introduced in [30] and [3] decompose trajectory data in the frequency domain, and amplify some features of the motion, such as depressed feelings or anger, by changing the coefficients of the basic functions.

Gleicher [12, 13] has succeeded in retargeting motion data to a character with a different body size using space-time constraints. In [21], hierarchical B-splines were used to edit/convert a motion to a similar motion with new constraints.

The advantage of kinematic methods lies in their simplicity. The algorithms are easy to implement, and the obtained motion is relatively natural for small changes.

There are also a number of motion-editing methods based on dynamics. Rose *et al.* [28] have extended the optimization method based on spacetime constraints to smoothly interpolate different captured-motion data.

Popovic´ *et al.* [26] have proposed a dynamics-based method to edit human motion data, which enabled animators not only to add constraints to the original motion, but also to change such physical parameters as the mass of the body segments or gravity which could not be done by previous kinematic methods. Their method is based on both PD control and spacetime constraints.

However, there has been little research directed to editing motion data using a precise human body model which incorporates the musculoskeletal system. Even though several researchers have used muscle models to obtain realistic rendering of human [29] or animal bodies [34], musculoskeletal models have rarely been used to yield or edit human motion data [18, 19]. Chen *et al.* [4] have created a very precise muscle model using FEM, but it has not been used for the control of human body models in a dynamic environment. Pandy *et al.* [25, 24] have used a musculoskeletal model to simulate maximum-height jumping. Their method is based on forward dynamics and optimal control. However, since no feedback controller is incorporated into this method, it is not suited to edit actual human motion, because the lack of a feedback controller makes it difficult to

handle motion such as a cyclic gait using forward dynamics.

3 Outline of the algorithm

In this paper, we propose a method to convert a captured motion dynamically and physiologically. For this purpose, a musculoskeletal human body model was prepared. The outline of the resultig algorithm is as follows:

1. The captured motion is retargetted to a similar motion that is feasible by the musculoskeletal human body model using our conversion algorithm that is explained in section 7.

2. Muscle parameters such as the effect of fatigue or the peak force that can be exerted by a muscle are changed to create a physiologically different motion. Dynamic parameters such as gravity or external force can also be changed to yield a physically different motion. To generate a tired motion, the history of the muscle activation level is calculated during the motion obtained in step 1, and then, by using the fatigue and recovery model, the physiological parameters which are used to calculate the maximum and minimum force exertable by a muscle are recalculated. To generate a motion by an injured body, the peak muscle force is defined by the animator.

3. Since the values for physical and physiological parameters have been changed, the motion obtained in step 1 is no longer feasible according to the musculoskeletal human body model. Therefore, the conversion algorithm is again applied to the motion to yield one that is feasible by the musculoskeletal model with the new physical parameters and muscle parameters. Thus, tired motion, motion by an injured body, or motion under a different physical enviornment can be obtained.

4 Musculoskeletal Model

In order to generate human motion based on the anatomy and physiology of a real human body, it is necessary to have a musculoskeletal human body model. The model used in our experiments has been generated from a number of sources.

For the legs, the model by Delp [7, 8, 9] has been used.[1] This data includes the attachment sites of 43 muscles on each leg and physiological parameters such as the length of tendons, range of joint angles, etc. The lower half of the body (Figure 4) is composed of the pelvis, and the femur, tibia, patella, talus, calcaneous, and toes in each leg. The joints of the legs are assumed as either a 3-DOF gimbal joint (hip joint) or as a 1-DOF joint (knee, ankle, and calcaneous joint). The data of Veeger and of Van Der Helm *et al.* [15, 14] have been used for the shoulder, and the data of Veeger *et al.* [27] has been used for the elbow. The upper half of the body (Figure 4) is composed of the torso, head, scapula, clavicle, humerus, ulna, radius, and the hand. The joints in the upper half of the body have been assumed to be either of 3-DOF gimbal type (torso, sternoclavicular and glenohumeral joint), 2-DOF type (acromioclavicular joint),

[1] The data can be downloaded from `http://isb.ri.ccf.org/isb/data/delp`

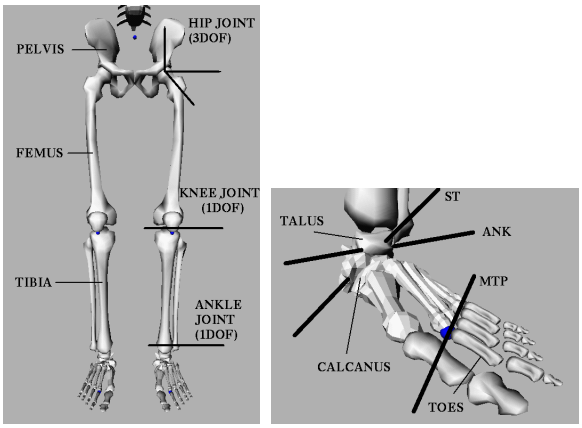


Figure 1: The rigid-body segments of the legs and joint axes. The ankle (ANK), subtalar (ST), and metatarsophalangeal (MTP) joints are modeled as pin joints with the axes shown

1-DOF type (humeroulnar), and sliding type (scapulothoracic gliding plane). There are 20 muscles on each arm, divided into a total 95 muscle elements.

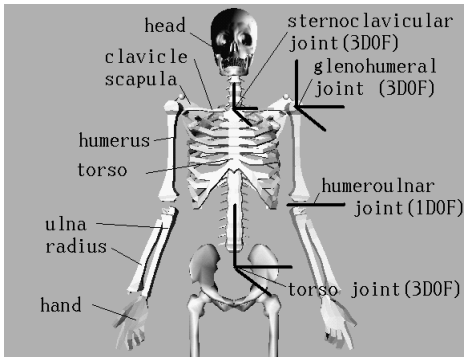
We removed some DOFs from the parameters for optimization, which were difficult to handle (such as the radio-ulnar and acromioclavicular joint, and the scapulothoracic gliding plane) and which were considered not important for motion conversion(head and hand joints), to ease the convergence of the criteria. As a result, the total DOF number is 34. Other necessary data, such as the weight and inertia of the segments were obtained from Yamaguchi *et al.* [38] for the legs, and from Veeger *et al.* [14] for the arms. The front and back views of the body model with muscles are shown in Figure 3.

5 The Muscle Model

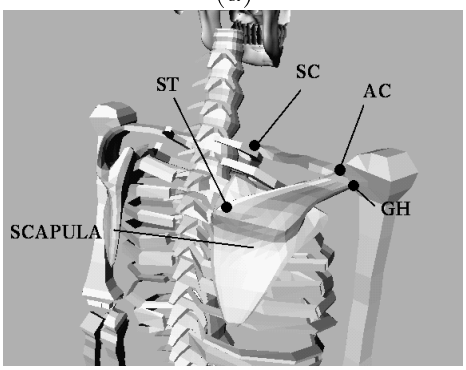
Each musculotendon is based on Hill's three component model (Figure 4). There are many muscle models which are derived from Hill's model [35], but the model used here is that of Delp *et al.* [7, 8]. The model is composed of three elements: the contractile element (CE, muscle fibers), the parallel elastic element (PEE, connective tissue around the fibers and fiber bundles), and the series elastic element (SEE, muscle tendon). The force exerted by each segment is denoted by f^{ce}, f^{pe} and f^T, and the length is denoted by l^{ce}, l^{pe} and l^T. The muscle-tendon length (l^{MT}) is the sum of the muscle fiber length and the tendon length:

$$l^{MT} = l^M \cos\alpha + l^T \tag{1}$$

where $l^M = l^{ce} = l^{pe}$. The muscle-tendon length (l^{MT}) can be calculated from the joint angles of each leg. The force exerted by the SEE (f^T) and PEE (f^{pe}) are functions only of their length (l^T, l^{pe}). There are four constant parameters which are specific to each muscle, their values being listed in [7, 8, 15, 14, 27]. These parameters are static ones which do not change during motion:



(a)



(b)

Figure 2: (a)The rigid-body segments and joints of the upper half of the body. (b)The backward view of the upper body. The articulations listed are the sternoclavicular joint (SC), the acromioclavicular joint (AC), the glenohumeral joint (GH), and the scapulothoracic gliding plane (ST).

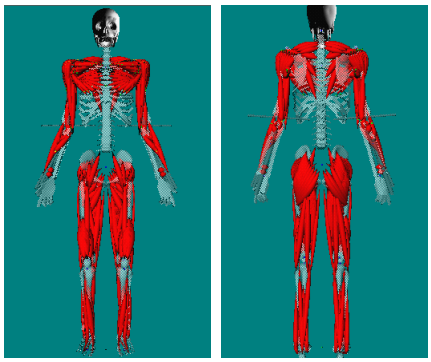


Figure 3: The frontal (left) and rear (right) views of the human body model

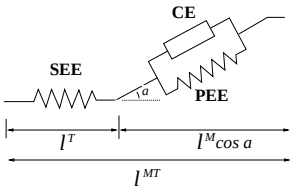


Figure 4: The muscle model by Hill used in this study

α pennation angle, the angle between the muscle and tendon
l_o^M the optimal length of the muscle fiber
f_o^M the maximal force value
l_s^T the tendon length when slack.

Variables called the normalized force and length are also defined for each element:

$$\begin{aligned}
\tilde{f}^{ce} &= \frac{f^{ce}}{f_o^M} \\
\tilde{f}^{pe} &= \frac{f^{pe}}{f_o^M} \\
\tilde{f}^{T} &= \frac{f^{T}}{f_o^M} \\
\tilde{l}^{M} &= \tilde{l}^{ce} = \tilde{l}^{pe} = \frac{l^M}{l_o^M} \\
\tilde{\epsilon}^{t} &= \frac{l^T - l_s^T}{l_s^T} \\
&= \frac{(l^{MT} - l^M cos\alpha) - l_s^T}{l_s^T} \\
&= \frac{(l^{MT} - \tilde{l}^M l_o^M cos\alpha) - l_s^T}{l_s^T}
\end{aligned}$$

The relationship between the force at each element is

$$f^T = (f^{ce} + f^{pe}) \cos\alpha \tag{2}$$

where f^{ce} depends on length l^{ce}, contraction velocity v^{CE}, and muscle activation level a which is controlled by the central nervous system (CNS).

A tendon is a passive element that exerts elastic force only when its length is greater than slack length l^{st}. The relationship between tendon strain ϵ^T and normalized tendon force $\tilde{f}^T$ is shown in Figure 5(a). CE can generate maximum force f_o^M when its length is the natural length (l_o^M) and the contraction velocity (v^{CE}) is zero. The relationships between $\tilde{f}^{ce}$ and $\tilde{l}^{ce}$ and between $\tilde{f}^{pe}$ - $\tilde{l}^{pe}$ are shown in Figure 5(b).

The force that can be exerted by CE decreases as the contraction velocity increases. The curve of $\tilde{f}^{ce}$ and v^{CE}/v^o where $a = 1$ and $l^{ce} = l_o^M$ is shown in Figure 5(c). v^o is the

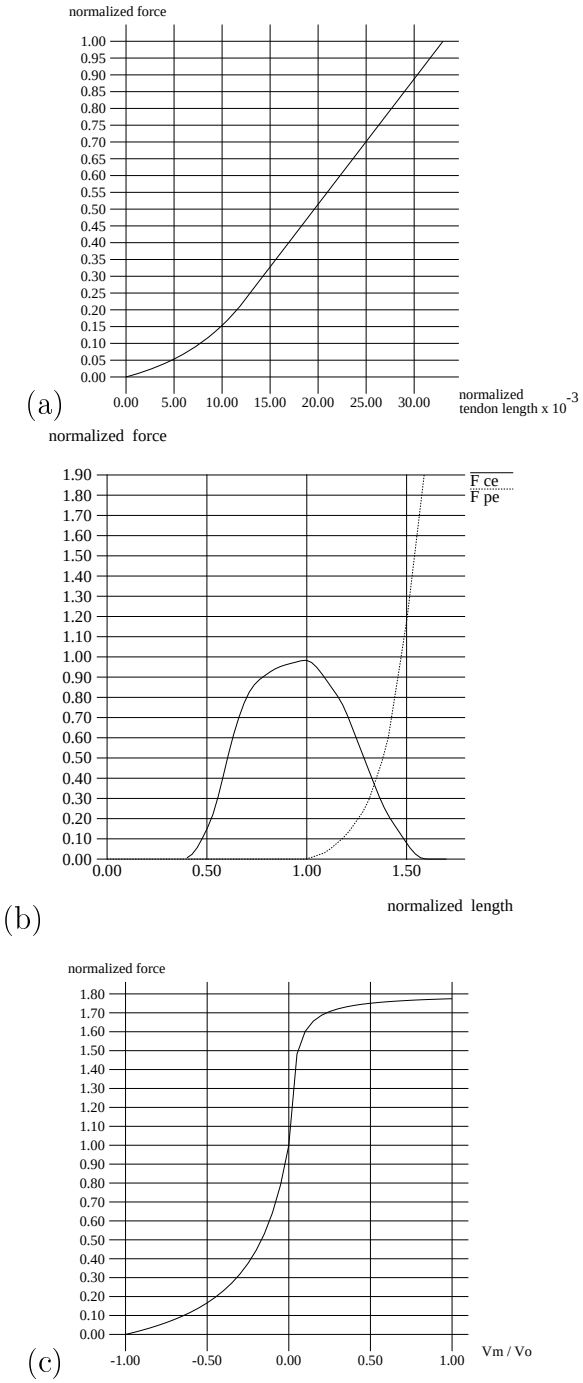


Figure 5: (a)Relationship between the tendon strain (ϵ^T) and the normalized tendon force $\tilde{(}f^T)$. (b) force-length curves for the active and passive muscle elements, $\tilde{f}^{ce}$ and their lengths, $\tilde{l}^{ce}$, $\tilde{f}^{pe}$ and $\tilde{l}^{pe}$. (c) Velocity-force curve of the contractile element when $l^M = l_o^M$ and $a = 1$. v^o is the maximal contraction velocity, which is assumed to be $10 l_o^M / s$ [25]. All data plotted from ref [7].

maximum contraction velocity by the CE and is assumed to be $10 l_o^M/s$[25]. This curve is defined as $g^{ce}(v^{CE}/v^o)$. Using $\tilde{f}^{ce}, \tilde{f}^{pe}, \tilde{f}^T$ and g^{ce}, we can obtain

$$f^{ce} = f_o^M \cdot \tilde{f}^{ce}(\tilde{l}^M) \cdot g^{ce}(v^m/v^o) \cdot a \quad (3)$$

$$f^{pe} = f_o^M \cdot \tilde{f}^{pe}(\tilde{l}^M) \quad (4)$$

$$f^T = f_o^M \cdot \tilde{f}^T(\tilde{\epsilon}). \quad (5)$$

From equation (2)-(5), we can derive:

$$\tilde{f}^T\left(\frac{(l^{MT} - \tilde{l}^M l_o^M cos\alpha) - l_s^T}{l_s^T}\right) -$$

$$\{\tilde{f}^{ce}(\tilde{l}^M) \cdot g^{ce}\left(\frac{v^m}{v^o}\right) \cdot a + \tilde{f}^{pe}(\tilde{l}^M)\} \cos\alpha = 0$$

while v^m can be calculated by the finite difference of the muscle length:

$$v^m = \frac{l^M - l_{prev}^M}{\Delta t}, \quad (6)$$

where l_{prev}^M is the length of l^M at the previous time step, and Δt is the length of the time step.

5.1 Calculating the Maximum and Minimum Forces that can be Exerted by the Muscles

Equation (6) can be used to calculate musculotendon force f^T when the activation level of the muscle is specified. When the activation level of the muscle is known, the only unknown variable in equation (6) is l^M. By letting the left term of equation (6) converge to 0, l^M can be calculated. l^T can be obtained using equation (1). Finally, the musculotendon force f^T can be calculated by equation (5).

A muscle exerts its maximum force when the muscle activation level a is 1 and the minimum force when a is 0. The maximum and minimum musculotendon force, f^{max} and f^{min}, can be computed by setting a to 0 and 1 in equation (6).

The musculotendon force, f^T, at each moment is limited by:

$$f^{min}(a = 0) \leq f^T \leq f^{max}(a = 1). \quad (7)$$

5.2 Fatigue model for the muscles

When a muscle exerts a large amount of force, fast glycolitic (FG) fibers in the muscle are recruited, which causes the intra-cellular pH level inside the muscle to decline. This causes the maximum amount of force exertable by the contractile element to decrease. This is called the *fatigue phase*. When the muscle is not used, the pH level increases, and the exertable force increases during the *recovery phase*.

Giat *et al.* [10] have observed the intra-cellular pH level inside an electronically stimulated quadriceps muscle using ^{31}P nuclear magnetic resonance spectroscopy and obtained the relationship between the intra-cellular pH level and force exerted by the muscle. Their fatigue and recovery model for muscles is explained in Appendix 3. Basically, the force exertable by a muscle's CE element changes according to how much force has been previously exerted.

6 Inverse Dynamics

After the trajectory of the body has been determined, the force and torque exerted at each joint can be calculated by specifying joint angle $\boldsymbol{\theta}$, angular velocity $\dot{\boldsymbol{\theta}}$, and angular acceleration $\ddot{\boldsymbol{\theta}}$. These operations are called inverse dynamics. We used a commercial software package based on Kane's method for our the calculations.

6.1 Solution of the Closed-loop Problem

When a human stands on two feet (the double support phase), the moment in the legs is redundant, and hence cannot be calculated using inverse dynamics. To overcome this problem, we distribute the force and moment to the legs in proportion to the inverse in the distance between the feet and the position on the ground where the body's center of gravity is projected [32, 17, 18].

6.2 Balance

To maintain the balance in the human body model, it is necessary to define a function that evaluates the stability of the posture. The 'zero moment point' (ZMP) can be used to define such a stability function. When a human stands, on one foot or on both of the feet, a point exists where the moment applied to the body from the ground is zero. When the body is supported by a single leg, this point is at the sole of the support foot, but when the body is supported by both legs, it is in an area surrounded by the feet [32]. Since there is no joint between each foot and the ground, the moment that can be generated between the sole and the ground is limited. If the moment exceeds that limit, the body will fall to the ground. One way to check this constraint is by calculating the ZMP of the support to see if it is within the support area.

If the ZMP is within the support area, the posture is stable. However, if the ZMP is outside the support area, an additional moment must be added to the support foot to prevent the body from falling down. In our model, the value of this additional moment is used to evaluate the postural stability:

$$\boldsymbol{s}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}) = \begin{cases} \boldsymbol{\tau}_+ & \text{(ZMP is outside of the support area)} \\ \mathbf{0} & \text{(ZMP is within the support area)} \end{cases} \tag{8}$$

where $\boldsymbol{\tau}_+$ is the minimum external moment that must be added to the support foot to achieve stability in the specified posture.

7 Conversion of human motion to a feasible simulation by using the musculoskeletal model

In this section, the algorithm for motion conversion using the musculoskeletal model is explained. This conversion algorithm is applied to the motion twice: when retargetting the original captured motion by the real human body to the musculoskeletal model, and when converting the motion to a feasible simulation by the musculoskeletal model after the muscle parameters have been changed.

The trajectories of the position of the body and the joint angles are represented by cubic B-spline functions

$$\theta_i = \sum_j c_i^j B^j \tag{9}$$

where θ_i is the kinematic trajectory of the ith-DOF, B^j are the basis B-spline functions, and c_i^j are their scalar coefficients.

Using the kinematic values, $\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}$, the joint moment can be calculated by inverse dynamics:

$$\boldsymbol{\tau} \;=\; \boldsymbol{f_1}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}). \tag{10}$$

It is necessary to evaluate at each moment whether each muscle can perform the motion in question. This evaluation is made by solving the following quadratic program:

$$\min_{\boldsymbol{f}, \boldsymbol{\tau}_{ext}} \|\boldsymbol{\tau}_{ext}\|^2$$
$$\text{subject to} \begin{cases} \boldsymbol{\tau} = \boldsymbol{A}\boldsymbol{f} - \boldsymbol{\tau}_{ext} \\ \boldsymbol{f}^{min} \leq \boldsymbol{f} \leq \boldsymbol{f}^{max} \end{cases}$$

where $\boldsymbol{f} = (f_1^T, ..., f_{n_m}^T)$, $\boldsymbol{f^{max}} = (f_1^{max}, ..., f_{n_m}^{max})$, $\boldsymbol{f^{min}} = (f_1^{min}, ..., f_{n_m}^{min})$, n_m is the number of muscles, $\boldsymbol{A}$ is the matrix that converts the muscle force to joint torque, $\boldsymbol{\tau}$ is the joint torque calculated by inverse dynamics using equation (10), and $\boldsymbol{\tau_{ext}}$ is the supplementary torque which is applied when the motion cannot be achieved only by the muscle force.

The human body model must also satisfy the balance constraint defined in Section 6.2. For the motion to be feasible, both $\|\boldsymbol{\tau}_{ext}\|^2 = 0$ and $\|\boldsymbol{s}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}})\|^2 = 0$ must be satisfied throughout the motion. Therefore, the feasibility of the motion is defined by

$$J = \int_{t_0}^{t_f} \|\boldsymbol{\tau}_{ext}\|^2 + \|\boldsymbol{s}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}})\|^2 dt. \tag{11}$$

Now, we calculate a physiologically feasible motion using optimization. The variables $\boldsymbol{x}$ in the optimization are the coefficients of the basis function and the time at which each motion terminates:

$$\boldsymbol{x} = (c_0^0, ..., c_m^0, ..., c_0^{n_c}, ..., c_m^{n_c}, t_f) \tag{12}$$

where m is the number of control points and n_c is the number of DOF of the system. The following problem is then solved to obtain a feasible motion:

$$\min_{\boldsymbol{x}} J \tag{13}$$

As J is minimized to zero, the state vector $\boldsymbol{x}$ gives a feasible motion by the musculoskeletal system.

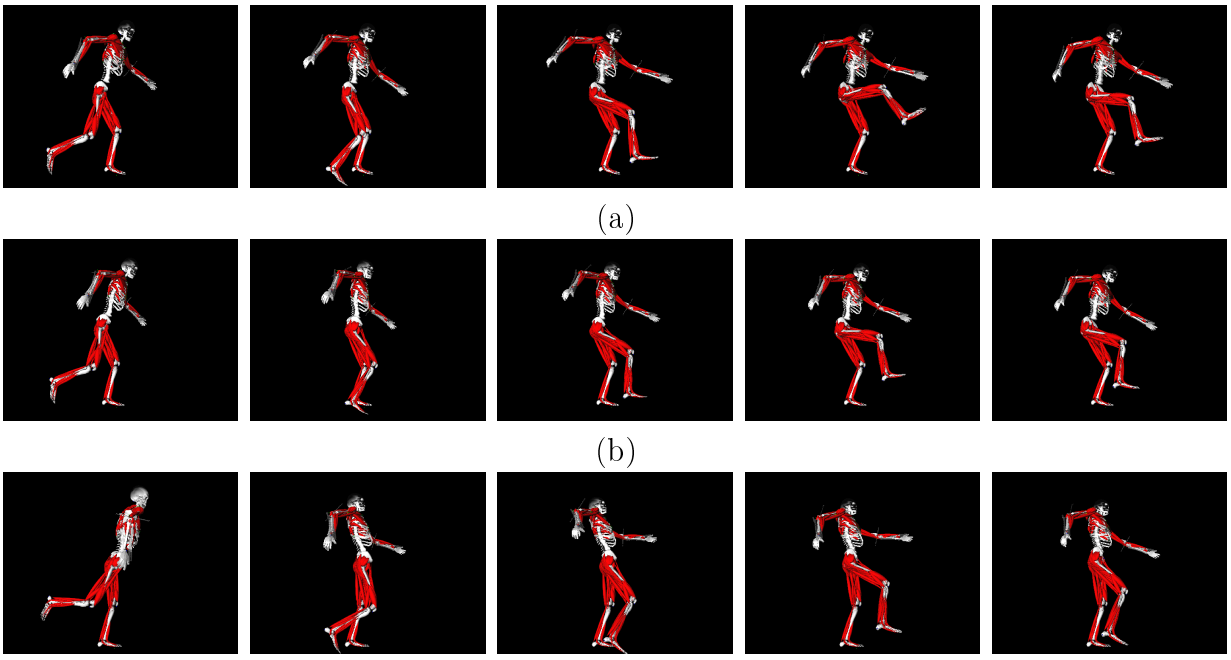
8 Experimental Results

In this section, a number of real human motions are physiologically converted by applying our method. The motions of the body were obtained by a magnetic motion capturing system. Unless the raw data were physiologically feasible, the conversion algorithm explained in the previous section was first applied to all the motions. After changing the muscle parameters or physical parameters, the conversion algorithm was applied to each motion again to obtain the final motion.

Two different kicking motions and a gait motion were first captured, and then new motions were obtained by changing the muscle parameters and dynamic parameters.

8.1 Kicking Motion

The first motion captured from the real human body is shown in Figure 8.1(a). Using inverse dynamics and the musculoskeltal model, the force exerted by each muscle was calculated throughout the motion. The pH level of each muscle was then calculated, and the peak muscle force by the contractile element was updated using the fatigue and recovery model. As the motion is repeated, what was previously feasible becomes infeasible, because of the decrease in the peak force by the contractile element of the muscle. Thus, the optimization problem shown in equation (11) is solved, to provide a feasible motion. This operation was repeated 50 times and the feasible motions after 20 and 50 kicks are shown in Figures 8.1(b) and(c). It is possible to observe the effect of fatigue in the lower foot position of the kicking leg at the final posture. As the kick is repeated, the sway of the upper part of the body increases, and therefore the motion becomes more unstable. Another example of a kicking motion is shown in Figure 8.1. This time, the original motion is a side kick as shown in Figure 8.1(a). Next, an elastic spring was attached to the ankle of the kicking leg, and the conversion algorithm was applied to the motion. As shown in Figure 8.1(b), the leg is pulled down by the spring. Therefore, the right leg passes close to the ground during the kicking motion.

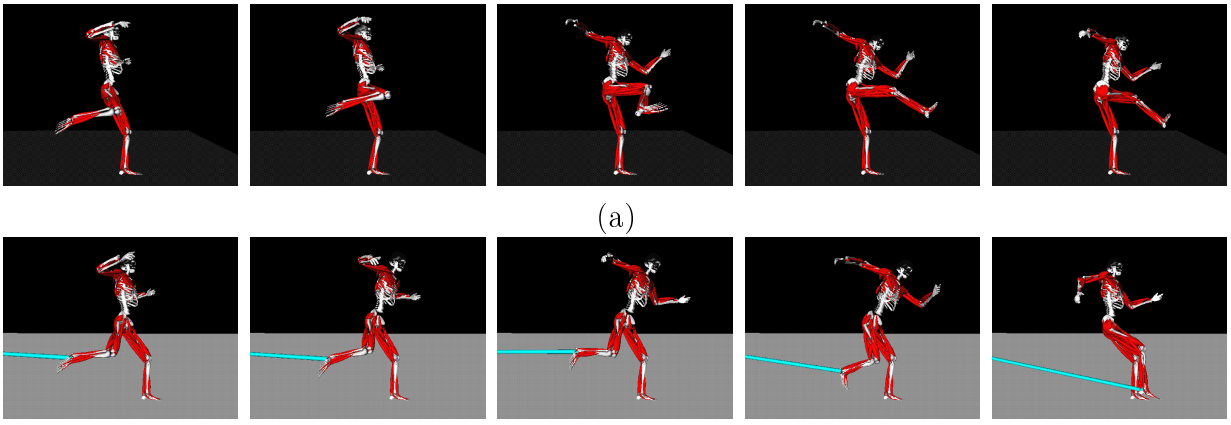


(a)

(b)

(c)

Figure 6: Conversion of the kicking motion: (a)initial motion, (b)converted motion after 20 repetitions, and (c) after 50 repetitions.



(a)

(b)

Figure 7: Conversion of the side kick: (a)initial motion and (b)converted motion.

8.2 Gait motion

A walking motion with a limp was created from an ordinary gait by reducing the maximum exertable force of the right muscles to $\frac{1}{5}$ of the original value and then optimizing equation (11). The initial gait motion and that with a limp are shown in Figures 8.1 (a) and (b). Next, a gait motion walking up a slope was created by changing the direction of the gravity. Starting the optimization from the motion in Figure 8.1 (a), a physiologically feasible slope-climbing gait was created as shown in Figure 8.1(c). The slope is 30 degrees. The footprints are set as constraints in these examples.

9 Discussion

Even though the use of a musculoskeletal system may seem complex and time-consuming, our method simplifies the process by avoiding the use of muscles as control parameters for optimization. In fact, the only extra computation necessary with our method, when compared with the traditional spacetime constraints problems, is the calculation of $\boldsymbol{A}$, $\boldsymbol{f^{max}}$ and $\boldsymbol{f^{min}}$ in equation (7), and of $\boldsymbol{\tau}_{ext}$ in equation (11). Even though the calculations for $\boldsymbol{A}$, $\boldsymbol{f^{max}}$ and $\boldsymbol{f^{min}}$ increase in proportion to the number of muscles, since each computation can be completed very quickly, the total time for their computation is not critical. In addition, since the solution for minimum $\boldsymbol{\tau}_{ext}$ is a quadratic programming problem, it can be solved very efficiently in a short time. As such, we consider the musculoskeletal system as more than just a special tool for specialized applications, it is also a tool that can be utilized in computer graphics for motion conversion.

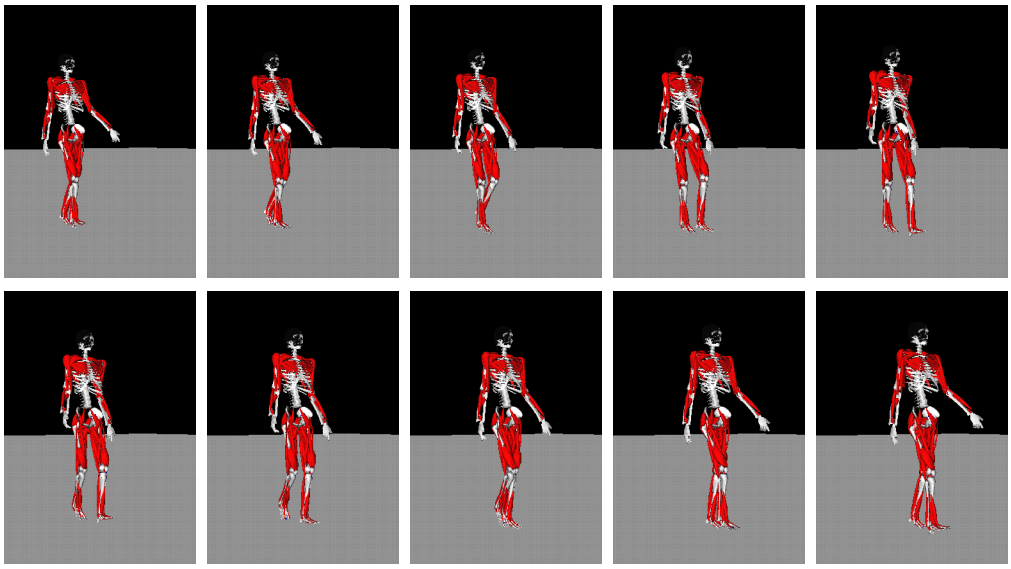
We believe that rather than computations required for muscle parameters, the more ctitical problem is the number of DOFs of the human body. In order to further develop the human body model precisely, more DOFs and attached muscles will be required. In order to handle this increase, it will be necessary to seek for an optimization algorithm that can handle many DOFs. One potential solution may be multiresolutional methods such as those proposed by Liu [23] and Lee [21].

10 Limitations

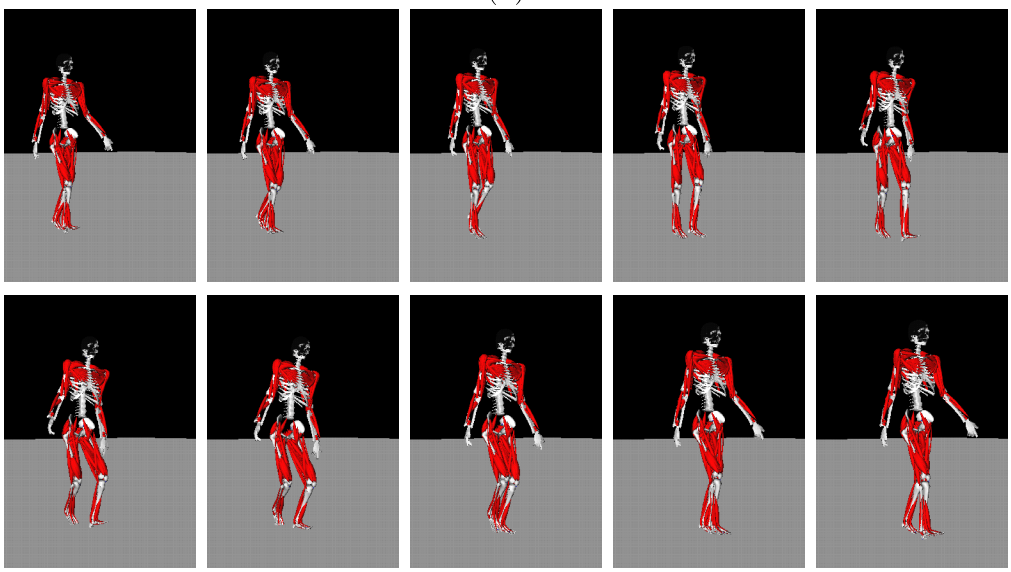
In this study, the torso has been greatly simplified. In particular, there are no muscles that connect the loins and the torso. In addition, the torso has only three degrees of freedom: the joints at the spine are all represented by one gimbal joint between the loins and the chest. In some cases, the chest joint oscillates in order to maintain the balance, and the motion does not look realistic. For example, in the kicking motion, as the the musculoskeletal model gets tired after repeating the kick, the chest part oscillates back and forth at a high frequency to maintain the balance. It will be necessary to model the spine more precisely to avoid this sort of motion.

11 Conclusions and Future Work

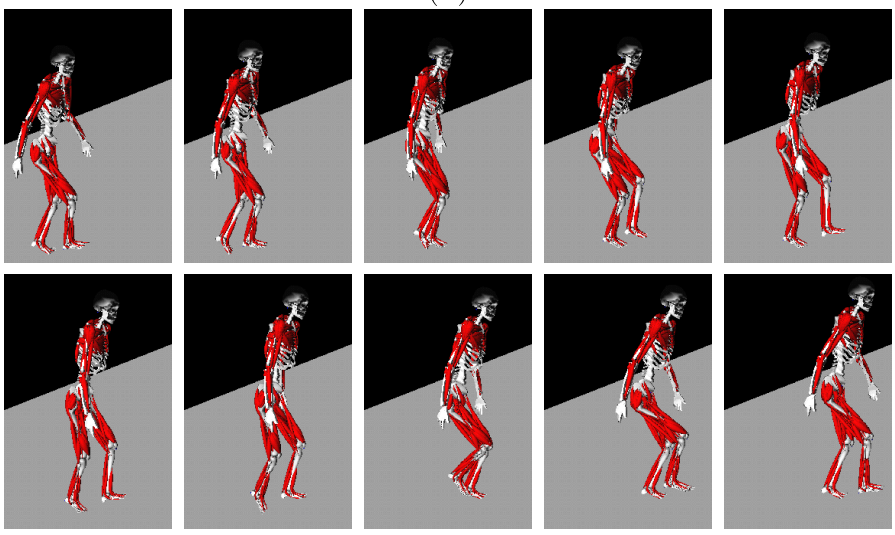
We have presented in this paper a method for physiologically converting real human motion data using a musculoskeletal model. We have shown how by combining the



(a)



(b)



(c)

Figure 8: Conversion of gait motion: (a) the initial motion, (b)limp gait created by weakening the muscles of the right leg, and (c) a gait over a 30 degrees steep slope.

musculoskeletal model with spacetime constraints, it is possible to create dynamically and physiologically appropriate and feasible motions. Our method makes it possible to simulate physiological effects such as fatigue and injury. This kind of simulation has been difficult with contemporary systems that have not taken into account the internal structure of the body. We have also shown the use of this system for motion editing by changing physical properties such as gravity.

In respect of future work, we have begun constructing a musculoskeletal model of the torso which will enable the generation and retargeting of a greater number of motions and provide greater realism. As well as applications in pure computer graphics, this method may also be useful for simulating rehabilitation. We also believe that retargeting the musculoskeletal model to various characters by adapting muscle parameters and physical values to different body sizes would be an interesting topic.

References

[1] W M Armstrong and M W Green. The dynamics of articulated rigid bodies for purposes of animation. *The Visual Computer*, 1:231–240, 1985.

[2] William W. Armstrong, Mark W. Green, and Robert Lake. Near-real-time control of human figure models. *IEEE computer graphics and applications*, June:52–61, 1987.

[3] A Bruderlin and L Williams. Motion signal processing. *Computer Graphics (Proceedings of SIGGRAPH 95)*, 29:97–104, 1995.

[4] D T Chen and D Zeltzer. Pump it up: Computer animation of a biomechanically based model of muscle using the finite element method. *Computer Graphics (Proceedings of SIGGRAPH 92)*, 26:89–98, 1992.

[5] M F Cohen. Interactive spacetime control for animation. *Computer Graphics (Proceedings of SIGGRAPH 92)*, 26:293–302, 1992.

[6] R D Crowninshield and R A Brand. A physiologically based criterion of muscle force prediction in locomotion. *Journal of Biomechanics*, 14:793–800, 1981.

[7] S Delp. *Surgery simulation: A computer graphics system to analyze and design musculoskeletal reconstructions of the lower limb.* PhD thesis, Stanford University, 1990.

[8] S Delp, P Loan, M Hoy, F E Zajac, S Fisher, and J Rosen. An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. *IEEE Transactions on Biomedical Engineering*, 37(8), August 1990. (Special issue on interaction with and visualization of biomedical data).

[9] SL. Delp and JP Loan. A graphics-based software system to develop and analyze models of musculoskeletal structures. *Computers in Biology and Medicine*, 25(1):21–34, 1995.

[10] Y Giat, J Mizrahi, and M Levy. A musculotendon model of the fatigue profiles of paralyzed quadriceps muscle under fes. *IEEE Transations on Biomedical Engineering*, 40(7):664–674, 1993.

[11] Y Giat, J Mizrahi, and M Levy. A model of fatigue and recovery in paraplegic's quadriceps muscle subjected to intermittent fes. *Transactions of the ASME: Journal of Biomechanical Engineering*, 118:357–366, 1996.

[12] M Gleicher. Retargetting motion to new characters. *Computer Graphicsi Proceedings (Proceedings of SIGGRAPH98)*, pages 33–42, 1998.

[13] M Gleicher and P Litwinowicz. Constraint-based motion adaptation. *The Journal of Visualization and Computer Animation*, 9:65–94, 1998.

[14] H.E.J.Veeger, F.C.T. Van Der Helm, L.H.V.Van Der Woude, G.M.Pronk, and R.H.Rozendal. Inertia and muscle contraction parameters for musculoskeletal modelling of the shoulder mechanism. *Journal of Biomechanics*, 24(7):615–629, 1991.

[15] F.C.T. Van Der Helm, H.E.J. Veeger, G.M.Pronk, L.H.V. Van Der Woude, and R.H.Rozendal. Geomoetry parameters for musculoskeletal modelling of the shoulder system. *Journal of Biomechanics*, 25(2):129–144, 1992.

[16] J K Hodgins, W L Wooten, D C Brogan, and J F O'Brien. Animation of human athletics. *Computer Graphics (Proceedings of SIGGRAPH 95)*, pages 71–78, 1995.

[17] H Ko and N I Badler. Animating human locomotion with inverse dynamics. *IEEE Computer Graphics and Applications*, March:50–59, 1996.

[18] T Komura, Y Shinagawa, and T L Kunii. Muscle-based feed-forward controller of the human body. *Computer Graphics Forum*, 16(3):C165–C176, 1997.

[19] Taku Komura, Yoshihisa Shinagawa, and Tosiyasu L. Kunii. Creating and retargetting motion by the musculoskeletal human body model. *The Visual Computer*, in press.

[20] J Laszlo, M van de Panne, and E Fiume. Limit cycle control and its application to the animation of balancing and walking. *Computer Graphics (Proceedings of SIGGRAPH 96)*, 30:155–162, 1996.

[21] J Lee and S Y Shin. A hierarchical approach to interactive motion editing for human-like figures. *Computer Graphics Proceedings (Proceedings of SIGGRAPH 99)*, pages 39–48, 1999.

[22] Richard L. Lieber. *Skeletal Muscle Structure and Function*, chapter 2, pages 95–100. Williams and Wilkins, 1992.

[23] Z Liu, S J Gortler, and M F Cohen. Hierarchical spacetime control". *Computer Graphics (Proceedings of SIGGRAPH 94)*, 28(2):35–42, 1994.

[24] M G Pandy, F C Anderson, and D G Hull. A parameter optimization approach for the optimal control of large-scale musculoskeletal systems. *Journal of Biomechanical Engineering : Transaction of the ASME*, 114:450–460, 1992.

[25] M G Pandy, F E Zajac, E Sim, and W S Levine. An optimal control model for maximum-height human jumping. *Journal of Biomechanics*, 23(12):1185–1198, 1990.

[26] Z Popovic and A Witkin. Physically based motion transformation. *Computer Graphics Proceedings (Proceedings of SIGGRAPH 99)*, pages 11–20, 1999.

[27] Veeger HE. Yu B. An KN. Rozendal RH. Parameters for modeling the upper extremity. *Journal of Biomechanics*, 30(6):647–652, 1997.

[28] C Rose, B Guenter, B Bodenheimer, and M F Cohen. Efficient generation of motion transitions using spacetime constraints. *Computer Graphics (Proceedings of SIGGRAPH 96)*, 30:147–152, 1996.

[29] F Scheepers, R E Parent, W E Carlson, and S F May. Anatomy-based modeling of the human musculature. *Computer Graphics (Proceedings of SIGGRAPH 97)*, pages 163–172, 1997.

[30] M Unuma, K Anjyo, and R Takeuchi. Fourier principles for emotion-based human figure animation. *Computer Graphics (Proceedings of SIGGRAPH 95)*, 29:91–96, 1995.

[31] M van de Panne. Parameterized gait synthesis. *IEEE Computer Graphics and Application*, March:40–49, 1996.

[32] M Vukobratović, B Borovac, D Surla, and D Stokić . *Biped Locomotion*. Springer-Verlag, 1990.

[33] J Wilhelms. Using dynamic analysis for realistic animation of articulated bodies. *IEEE Computer Graphics and Application*, June:13–27, 1987.

[34] J Wilhelms and A V Gelder. Anatomically based modeling. *Computer Graphics (Proceedings of SIGGRAPH 97)*, pages 172–180, 1997.

[35] J M Winters. Hill-based muscle models: A systems engineering perspective. In Jack M. Winters and Savio L-Y. Woo, editors, *Multiple muscle systems: Biomechanics and movement organization*, chapter 5, pages 69–93. Springer-Verlag, 1990.

[36] A Witkin and M Kass. Spacetime constraints. *Computer Graphics (Proceedings of SIGGRAPH 88)*, 22:159–168, 1988.

[37] W L Wooten and J K Hodgins. Animation of human diving. *Computer Graphics Forum*, 15(1):3–13, 1996.

[38] D G Yamaguchi and F E Zajac. Restoring unassisted natural gait to paraplegics via functional neuromuscular stimulation : a computer simulation study. *IEEE Transactions on biomedical engineering*, 37:886–902, 1990.

[39] Jianmin Zhao and Norman I. Badler. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Transactions on Graphics*, 13(4):313–336, 1994.

Appendix 1: Muscle Force Prediction

In order to calculate the amount of fatigue for each muscle, it is necessary to predict the muscle force during the motion. At each moment, the relationship between the joint torque and muscle force is linear. The torque $\tau_\mathbf{i}$ exerted at joint i, is generated by the muscles crossing the joint:

$$\tau_\mathbf{i} = \sum_j \mathbf{r_j} \times \mathbf{f_j} \tag{14}$$

where $\mathbf{r_j}$ and $\mathbf{f_j}$ are the moment arm and the force exerted by muscle j, respectively, and $\times$ represents the outer product. The moment arm $\mathbf{r_j}$ can be easily calculated from the muscle attachment sites and joint angles, and $\tau_\mathbf{i}$ can be calculated using inverse dynamics. However, since the number of muscles crossing joint i is always greater than the degree of freedom of the joint, solving f_j in equation (14) is a redundant problem and an optimization method is applied to determine the muscle force.

Crowninshield et al. [6] proposed a criteria which is based on the inversely nonlinear relationship of muscle force and contraction endurance. Their criteria u were written in the form

$$u = \sum_{i=1}^{m} (\frac{f_i^T}{A_i})^n \tag{15}$$

where $f_i^T = |\mathbf{f_i}|$, A_i is the PCSA of muscle i and n is 1, 2, 3 or 100. They compared the calculation results with electromyographic data and reported $n = 2$ provided good results. In this research $n = 2$ has been used.

Using equations(14) as equality constraints and equations (7) as inequality constraints, u can be minimized by quadratic programming. The muscle force f^T at a given moment can also be obtained. Thus l^M is calculated using f^T, and l^{MT}. This l^M is used as l^M_{prev} in the next stage to calculate v^m by finite differentiation, and then f^{min} and f^{max} can be calculated again at the next stage. By forward repetition of this calculation, it is possible to calculate f^{min} and f^{max} at any stage during the motion.

Appendix 2: Calculating the Muscle Activation Level

Since our method is based on spacetime constraints, we need to calculate various muscle factors from the trajectories of the body. In the beginning, each musculotendon length l_t^M can be calculated kinematically.

When the musculotendon force f^T is given, it is possible to calculate the muscle activation level using equations (2)-(4). First, using equation (5), the muscle tendon length l^T can be obtained. Then, l^M can be calculated from equation (1). As l^M is known, f^{pe} is obtained from equation (4), and since f^T is already given, f^{ce} is calculated using equation (2). Finally, the activation level a can be calculated from equation (3).

Appendix 3: Muscle Fatigue and Recovery Model

In this section, the fatigue and recovery model proposed by Giat et al. [10, 11] is explained. This model can be used to determine the maximal amount of force exertable by a contractile element as time passes.

The decay of the pH level during the fatigue phase with time t is calculated by

$$pH^F(t) = c_1 - c_2 tanh[c_3(t - c_4)] \tag{16}$$

with constant parameters c_1, c_2, c_3 and c_4.

The pH level during the recovery phase is similarly calculated by

$$pH^R(t) = d_1 + d_2 tanh[d_3(t - d_4)] \tag{17}$$

with constant parameters d_1, d_2, d_3 and d_4. The force output is represented by the following function:

$$f_{pH}(pH) = d_5[1 - e^{d_6(pH - d_7)}]. \tag{18}$$

where d_5, d_6 and d_7 are constant values. Equation (18) is normalized by the force obtained at the beginning of the experiment:

$$f^N_{pH}(pH) = \frac{f_{pH}(pH(t))}{f_{pH}(pH(t_0))} \tag{19}$$

where $0 < f^N_{pH} < 1$. The values of the constant parameters defined here are listed in Table 1.

Function	Parameter	Value
pH_F	c_1	6.70
	c_2	0.502
	c_3	0.0406
	c_4	30.0
pH_R	d_1	6.55
	d_2	0.502
	d_3	0.0026
	d_4	475.806
f_{pH}	d_5	1136.
	d_6	-0.0097
	d_7	6.0934

Table 1: Parameter values for Giat's fatigue and recovery model

The normalized force-pH function $f^N_{pH}(pH)$ is combined with equation (3), to compose a new dynamic equation of the CE component:

$$f^{ce} = f^{ce}(l^{ce}, v^{CE}, a) \cdot f^N_{pH}. \tag{20}$$

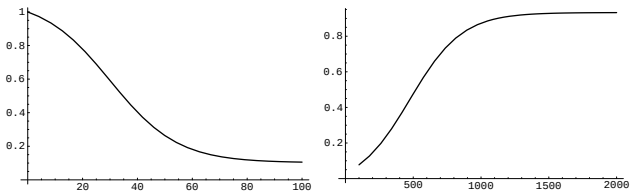


Figure 9: The relationship between time (seconds) and the normalized force by a muscle on fatigue (left) and recovery phase (right)

The decay and recovery of the normalized force during the fatigue and recovery phase is shown in figure 11. The problem with this model is that the relationship between the activation level and the pH derivation is not included. Previous experiments have shown that the FG fibers begin to be recruited when the activation level is above 50% of maximal contraction [22]. Therefore, in this research, the phase of each muscle is switched between fatigue and recovery one according to its activation level. The threshold is set to 0.5.